CONSENSUS CONTACT PREDICTION BY LINEAR PROGRAMMING

Xin Gao[1], Dongbo Bu[1,2*], Shuai Cheng Li[1], Ming Li[1†], and Jinbo Xu[3‡]
[1] *David R. Cheriton School of Computer Science*
University of Waterloo, Waterloo
Ontario, Canada N2L 3G1
[2] *Institute of Computing Technology*
Chinese Academy of Sciences, Beijing, China 100080
[3] *Toyota Technological Institute at Chicago*
1427 East 60th Street, Chicago, IL 60637
Email: {x4gao, dbu, scli, mli}@cs.uwaterloo.ca, j3xu@tti-c.org

Protein inter-residue contacts are of great use for protein structure determination or prediction. Recent CASP events have shown that a few accurately predicted contacts can help improve both computational efficiency and prediction accuracy of the *ab inito* folding methods. This paper develops an integer linear programming (ILP) method for consensus-based contact prediction. In contrast to the simple "majority voting" method assuming that all the individual servers are equal and independent, our method evaluates their correlations using the maximum likelihood method and constructs some latent independent servers using the principal component analysis technique. Then, we use an integer linear programming model to assign weights to these latent servers in order to maximize the deviation between the correct contacts and incorrect ones; our consensus prediction server is the weighted combination of these latent servers. In addition to the consensus information, our method also uses server-independent correlated mutation (CM) as one of the prediction features. Experimental results demonstrate that our contact prediction server performs better than the "majority voting" method. The accuracy of our method for the top $L/5$ contacts on CASP7 targets is 73.41%, which is much higher than previously reported studies. On the 16 free modeling (FM) targets, our method achieves an accuracy of 37.21%.

Keywords: residue-reside contact prediction, consensus, principal component analysis, integer linear programming, latent server.

1. INTRODUCTION

Computational protein structure prediction has made great progress in the last three decades [1, 2]. Recent CASPs (Critical Assessment of Structure Prediction) [3–8] have demonstrated that accurately predicted contacts can provide very important information for protein structure prediction methods. Rosetta [9–11] performs impressively on recent CASPs. Misura *et al.* [12] further modified the Rosetta free modeling protocol to encode residue-residue contact information. Experimental results demonstrate that by using spatial constraints extracted from homologous structures, not only the running time is shortened, but also the prediction accuracy is improved. For some concrete cases, the models built by Rosetta are more accurate than their templates on aligned regions, which is rarely seen before. Zhang-server, a refined version of TASSER [13], ranked number 1 among all the automatic servers in CASP7. CASP7 evaluation shows that iteratively running TASSER simulation for two rounds by using contact constraints at the second round greatly improves the prediction accuracy.

For a protein of length L. The contact map of this protein is an $L \times L$ matrix A, in which $A[i][j]$ is set to 1 if residue i and j are in contact, and 0 otherwise. Commonly, two residues are considered to be in physical contact if the spatial distance between their C_β atoms (C_α atom for Glycine) is less than some threshold value.

1.1. Related Work

There are four commonly acknowledged contact prediction assessment criteria: accuracy, coverage, improvement over random, and X_d [7, 8, 14]. Among them, accuracy is the most important measurement.

Protein residue-residue contact map was first studied by [15–18] to calculate mean force potentials.

*The first two authors contribute equally to the paper.
†To whom correspondence should be addressed.
‡To whom correspondence should be addressed.

Göbel *et al.* [19] formally proposed the problem of residue-residue contact prediction, and showed that correlated mutation (CM) is useful information to predict inter-residue contacts. Different correlated mutation calculation methods have been carefully studied since then [20–23].

While CM performs well with local contact prediction, which is usually considered to be two in-contact residues within 6 amino acids from each other on protein sequence, it usually fails on non-local contacts. Therefore, other information such as evolutionary information and secondary structure information, has been applied to improve the performance of contact prediction methods [24–30]. In [24], Fariselli *et al.* encoded four kinds of features into a neural network based server (CORNET): 1) CM, 2) evolutionary information, 3) sequence conservation, and 4) predicted secondary structure. They defined two residues to be in contact if the Euclidean distance between the coordinates of their C_β atoms (C_α atom for Glycine) is smaller than 8Å. To fairly test the performance of CORNET, they further required that the sequence separation between residues is at least 7, which can eliminate the influence of local α-helical residue-residue contacts. CORNET has an average accuracy of 0.21, which is higher than any previously reported result. Other features have been well studied since then [26, 28]. However, the reported accuracy has not been improved too much. PROFcon [30], one of the top three contact prediction servers in CASP6 [7], encodes alignment information into their neural network model, such as solvent accessibility and secondary structure over regions between two residues, as well as the average properties of the entire protein. PROFcon performs impressively on short proteins or alpha/beta proteins, on which the accuracy is over 30%.

Different from those machine learning based methods, which encode CM information and other sequence-related and alignment-related information, there are some studies trying to predict residue-residue contacts from other perspectives. Bystroff and his colleagues [31, 32] took folding pathway into consideration, and predicted residue contacts by HMMSTR [33], a hidden Markov model for local sequence-structure correlation. MacCallum [34] first pre-processed the sequence profile generated by PSI-BLAST [35]. Then Self-Organizing Maps (SOMs) were applied to reduce the high dimension of the profile data to 3D SOM grids. When converting into RGB code, contacted β-strands usually have correlated colors.

To sum up, previous studies have drawn the following conclusions: 1) Correlated mutation information is an influential factor in contact prediction, while solely encoding CM is not good enough for predicting contacts; 2) Other information, such as secondary structure, and solvent accessibility can help improve the accuracy; 3) Contacts predicted by top protein structure prediction servers are comparable or even a bit better than those predicted by contact predictors.

1.2. Our Contributions

To take advantage of useful information from the above conclusions, we propose a consensus residue-residue contact prediction method. Our consensus method assigns a confidence score to each contact from all contacts predicted by individual protein structure prediction servers, while also taking CM information into consideration. The intuition behind our method is that top models generated by protein structure prediction servers are usually the results of optimization on global energy and structures. Thus, encoding such information can help to select conserved contacts and long-ranged contacts. Different from traditional consensus methods which are widely used in protein fold recognition, our method aims to be able to identify correctly predicted contacts even if the majority of servers votes against them.

We have observed from recent CASP results that correlation exists among different servers on contacts determined by predicted 3D models because of similar tools used by these servers, such as PSI-BLAST and PSIPRED [36]. The correlated servers sometimes make a native contact to receive less supports than some incorrect ones. Our consensus method aims to reduce the impact caused by server correlation. The outline of our consensus method is as follows:

- A maximum likelihood (ML) method is applied to measure correlation coefficient between any two servers.
- Principal component analysis (PCA) technique is employed to extract new independent latent servers.
- An integer linear programming(ILP) method is then used to assign a weight

to each latent server, by maximizing the confidence score difference between native contacts and incorrect ones. CM is also considered to be a latent server which assigns a probability to each contact. This results in a consensus contact predictor to accurately assign confidence scores for all contacts extracted from the initial models.

The rest of this paper is organized as follows: Section 2 presents some preliminaries. In Section 3, we describe our new consensus method. Section 4 shows and analyzes experimental results on CASP7 data set. In Section 5, we discuss the potential applications and the future development of our method. Finally, Section 6 draws some conclusions.

2. PRELIMINARIES

In this paper, a *model* refers to a protein structure outputted by a protein structure prediction server. In contrast to human expert, a *server* refers to an automated system which predicts a set of structures for a given amino acid sequence, known as the *target*. Two residues are in contact if their C_β atoms (C_α atom for Glycine) is smaller than 8Å and they are at least 6 residues apart in the sequence. We call a contact *native contact* if the two residues are indeed in contact in the native structure of the target.

Given a model and a target, the contact accuracy of this model is calculated as the number of native contacts extracted from this model divided by the total number of contacts of this model, while the contact coverage of this model is defined to be the number of native contacts extracted from this model divided by the total number of native contacts. Since contacts extracted from protein structure prediction servers do not have confidence scores, we randomly choose a number of contacts to do statistics, for example, L, $L/5$ or all, where L is the length of the target protein.

Given a target t_l, $1 \leq l \leq \ell$, a server S_i, $1 \leq i \leq u$, outputs a set of models. The contacts determined by these models are extracted and considered as contact candidates, denoted as $C_{i,l} = \{c_{i,l,q} | 1 \leq q \leq n_{i,l}\}$, where $n_{i,l}$ is the number of contacts produced by server S_i for target t_l. The set of contact candidates for target t_l is denoted as $C_l = \bigcup_i C_{i,l}$. A consensus server aims to assign a confidence score to each candidate.

This paper is based on the following two assumptions:

- Server S_i generates its predictions based on a confidence measure. That is, for each contact $c \in C_l$, S_i has a confidence $s_{i,c,l}$ that c appears in the native structure. Since the initial confidence score is unavailable, we simply approximate it by the number of models containing this contact divided by the total number of models generated by the server on this target.
- There are some implicit latent independent servers H_j, $1 \leq j \leq v$, dominating the explicit servers S_i. Given a target t_l, H_j assigns a value $h_{j,c,l}$, $c \in C_l$, as the confidence that c is a native contact.

Identifying the latent independent servers is essential to reduce the negative effects of server correlations and to reduce the dimensionality of the search space, as the number of latent servers is expected to be smaller than the number of original servers. After deriving the latent servers, we can design a new and more accurate prediction server S^*, by an optimal linear combination of the latent servers, which for each target t_l assigns a confidence score to each contact candidate $c \in C_l$ as follows:

$$s^*_{l,c} = \sum_{j=1}^{v} \lambda^*_j h_{j,c,l} \tag{1}$$

where λ^*_j is the weight of latent server H_j.

3. METHODS

The basic idea of our method is to reduce the negative effects caused by the correlations among prediction servers. We first employ the maximum likelihood technique to estimate the server correlations; then adopt the factor analysis technique to uncover the latent servers; and finally design a mixed integer linear programming method to derive the optimal weights for the latent independent servers.

3.1. Maximum Likelihood Estimation of Server Correlations

Let $O_{i,j,l}$ denote the overlap set of $C_{i,l}$ and $C_{j,l}$, i.e., $O_{i,j,l} = C_{i,l} \cap C_{j,l}$, and let $o_{i,j,l} = |O_{i,j,l}|$. For a given

target, let $p_{i,j}$ be the probability that a contact returned by S_i is the same to that returned by server S_j. Under a reasonable assumption that targets t_l, $1 \leq l \leq \ell$ are mutually independent, the likelihood that server S_i, $1 \leq i \leq u$ generates contacts $c_{i,l,q}$, $1 \leq q \leq n_{i,l}$ is:

$$L(p_{i,j}) = \prod_{l=1}^{\ell} \binom{n_{i,l}}{o_{i,j,l}} p_{i,j}^{o_{i,j,l}} (1-p_{i,j})^{n_{i,l}-o_{i,j,l}} \quad (2)$$

Therefore, the maximum likelihood estimation of $p_{i,j}$ can be calculated as follows:

$$p_{i,j} = \frac{\sum_{l=1}^{\ell} o_{i,j,l}}{\sum_{l=1}^{\ell} n_{i,l}} \quad (3)$$

In the rest of this paper, we use P to denote the matrix $P = [p_{i,j}]_{u \times u}$.

3.2. Uncovering the Latent Servers

For a target t_l, let $s_{i,c,l}$ and $h_{j,c,l}$ be the confidence that contact c is chosen as one of the prediction results by server S_i and H_j, respectively. Since the latent servers are mutually independent, it is reasonable to assume that $s_{i,k,l}$ is a linear combination of $h_{j,k,l}, 1 \leq j \leq v$:

$$\overrightarrow{s_{i,l}} = \sum_{j=1}^{v} \lambda_{i,j} \overrightarrow{h_{j,l}}, \qquad \sum_{j=1}^{v} \lambda_{i,j} = 1, \qquad 1 \leq i \leq u. \quad (4)$$

where $\overrightarrow{s_{i,l}} =< s_{i,1,l}, s_{i,2,l}, \ldots, s_{i,|C_l|,l} >$, $1 \leq i \leq u$, and $\overrightarrow{h_{j,l}} =< h_{j,1,l}, h_{j,2,l}, \ldots, h_{j,|C_l|,l} >, 1 \leq j \leq v$. Here, $\lambda_{i,j}$ is the weight, and a larger $\lambda_{i,j}$ implies a higher chance that server S_i adopts contacts reported by H_j.

From the correlation matrix of prediction servers S_i, factor analysis technique is employed to derive $\lambda_{i,j}$ and $\overrightarrow{h_{j,l}}$; that is, $\overrightarrow{h_{j,l}}$ can be represented to be a linear combination of $\overrightarrow{s_{i,l}}$ as follows:

$$\overrightarrow{h_{j,l}} = \sum_{i=1}^{u} \omega_{j,i} \overrightarrow{s_{j,i,l}}, \qquad 1 \leq j \leq v, \qquad 1 \leq l \leq \ell \quad (5)$$

where $< \omega_{j,1}, \omega_{j,2}, \cdots, \omega_{j,n} >$ is an eigenvector of $P^T P$.

3.3. ILP Model to Weigh Latent Servers

After deriving the latent servers $H_j (1 \leq j \leq v)$, we can construct a new server S^*, as an optimal linear combination of the latent servers. For each target t_l, it assigns each contact candidate $c \in C_l$ with a score as in Eq 1.

To determine a reasonable setting of coefficient λ_k^*, a training process is conducted on a training data set $D = \{< t_l, C_l^+, C_l^- >, 1 \leq l \leq |D|\}$, where $t_l \in T$ is a target, $C_l^+ \subseteq C_l$ denotes the set consisting of native contacts, and $C_l^- \subseteq C_l$ denotes the incorrect contact set. The learning process attempts to maximize the number of contacts that are correctly identified by S^*.

More specifically, for each target t_l in the training data set, a score is assigned for each contact candidate by S^*. A good contact predictor should assign native contacts higher scores than incorrect ones. The larger the gap between scores of native contacts and incorrect ones, the more robust this new prediction approach is. In practice, "soft margin" idea is adopted to take outliers into accounts; that is, allowing errors on some samples, we maximize the number of native contacts with a score higher than incorrect ones by at least a threshold.

In our integer linear programming formulation, we employ two types of indicator variables. Let $x_{p,q}$ be an integer variable such that $x_{p,q} = 0$ if and only if contact p is given score higher than q by at least ϵ. Here, ϵ is a parameter used as the lower bound of gap between the score of a native contact and incorrect ones. Similarly, let $y_{p,l}$ denote whether p has a score greater than all the incorrect contacts in C_l^-.

Formally, the learning techniques can be formulated into an ILP problem as follows:

$$max \sum_{l=1}^{|D|} \sum_{p \in C_l^+} y_{p,l} \quad (6)$$

$$subj. \quad to \quad \sum_{j=1}^{v} \lambda_j^* H_{j,p,l} - \sum_{j=1}^{v} \lambda_j^* H_{j,q,l}^* - \epsilon \geq x_{p,q} - 1$$

$$p \in C_l^+ q \in C_l^-, 1 \leq l \leq |D| \quad (7)$$

$$\frac{1}{|C_l^-|} \sum_{q \in C_l^-} x_{p,q} \geq y_{p,l} \quad p \in C_l^+, 1 \leq l \leq |D| \quad (8)$$

$$\sum_{j=1}^{v} \lambda_j^* = 1, \lambda_j \geq 0 \quad 1 \leq j \leq v \quad (9)$$

$$x_{p,q} \in \{0,1\} \quad y_{p,l} \in \{0,1\} \tag{10}$$

For constraint 7, it is easy to see that $\sum_{j=1}^{v} \lambda_j^* H_{j,p,l} - \sum_{j=1}^{v} \lambda_j^* H_{j,q,l}^* \geq -1$. Thus, this constraint forces $x_{p,q}$ to be 0 if the difference between the scores assigned to p and q is smaller than ϵ. If p has a score not higher than all the incorrect contacts, constraint 8 will force $y_{p,l}$ to be 0. Constraint 9 normalizes the coefficient settings, and constraint 10 restrict $x_{p,q}$ and $y_{p,l}$ to be either 0 or 1. The objective function is the number of native contacts with score higher than all the incorrect contacts.

3.4. A New Prediction Server

Now, we wrap up everything to obtain a new prediction server. Given a target t^*, each server S_i produces a set of contact candidates, C_i^*. The set of all candidates is denoted as $C^* = \bigcup_i C_i^*$. For each contact candidate $c \in C^*$, the latent probability $h_{j,c}^* = \sum_{i=1}^{u} \omega_{j,i} s_{i,c}^*$, $1 \leq j \leq v$, is derived from Eq. 5. Then, the consensus server produces a score for each contact candidate based on Eq. 1, and picks up the top scored ones as the final predictions.

4. EXPERIMENTAL RESULTS

4.1. Data Set

Server Selection. To fairly evaluate the performance of our consensus method, we chose six automatically individual protein structure prediction servers, each of which is comparative modeling method. These servers are FOLDpro [37], mGenThreader [38, 39], RAPTOR [40, 41], FUGUE3 [42], SAM-T02 [43], and SPARK3 [44]. Although there are some fragment assembly based servers with higher overall performance on protein 3D structure prediction than these six servers, such as Rosetta and Zhang-server, we didn't choose them because their assembling process directly uses the results of some contact prediction methods.

Training and Test Data. The biennial CASP has provided us a comprehensive and objective data set. We chose CASP7 targets and models generated by those six servers as our training and test data. For each server on a target, top 5 models are considered. All server models are downloaded from CASP7 website, except for mGenThreader, which did not participate CASP7. We submitted CASP7 targets to mGenThreader web server and downloaded models from there. Eighty-nine CASP7 target proteins have their native structures published after the CASP7 while 104 protein sequences were released as targets. We removed redundancy at 40% sequence identity level using CD-HIT [45], which results in a data set with 88 target proteins. Only T0346 is removed because it shares 71% sequence identity with T0290. We further removed three targets (T0287, T0334, and T0385) from our data set because there are some errors in models generated by some of the six individual servers. To do cross validation, we randomly divided the 85 target proteins into four sets with size 22, 21, 20, and 22, respectively. If one target belongs to a set, then all of its models and contacts are in this set.

Data Set Statistics. We compared the performance of the six individual servers from the contact prediction accuracy and coverage point of view. The prediction accuracy of a server is calculated as the number of correctly predicted contacts divided by the total number of predicted contacts by this server, while the coverage of a server is calculated as the number of correctly predicted contacts divided by the total number of real contacts in the native structure. For each server on each target, the best model among the top 5 models generated by this server in terms of contact accuracy is chosen. If the number of contacts generated by a model is less than $L/5$, both the accuracy and the coverage for this model are set to 0. As shown in Table 1, the average accuracy among all contacts determined by the best model ranges from 43% to 53%, while the SAM-T02 server has the highest accuracy. The server "Overall" in Table 1 means the server which contains all contacts determined by the best six models generated by those six servers. The average accuracy of server "Overall" is very low (12.30%) comparing to the average accuracy of any individual server. Recall the way to calculate the accuracy, the server "Overall" always contains much more correctly predicted contacts than any individual server does. Therefore, the low accuracy of server "Overall" implies the incorrectly predicted contacts generated by these individual servers are different from each other for most cases, which means consensus method can probably be applied to differentiate correctly predicted contacts and incorrectly predicted ones.

It can also be seen that the average coverage

of these six servers ranges from 36% to 51%, while RAPTOR has the highest coverage. However, when combining these six servers together, the average coverage for server "Overall" is very high (about 80%). This means some correctly predicted contacts are only supported by a small number of individual servers while different servers can predict a common subset of correctly predicted contacts.

Note that to fairly evaluate the contact prediction ability of a protein structure prediction server, both accuracy and coverage should be combined. For example, SAM-T02 generates the highest accuracy among the six individual servers. However, the coverage is low (37.1%). This reveals that SAM-T02 tends to generate protein structure models which contain only a small number of contacts, most of which are conserved contacts.

Table 1. The average and deviation of contact accuracy and coverage of the best model among the top 5 models generated by different individual servers on CASP7 targets.

Server	Ave_{Accu}	Dev_{Accu}	Ave_{Cov}	Dev_{Cov}
FOLDpro	0.4511	0.0818	0.4836	0.0928
mGenThreader	0.4317	0.0659	0.4480	0.0851
RAPTOR	0.4843	0.0664	0.5221	0.0697
FUGUE3	0.4630	0.0793	0.3667	0.0554
SAM-T02	0.5331	0.0651	0.3710	0.0551
SPARK3	0.4793	0.0731	0.5118	0.0764
Overall	0.1230	0.0072	0.8028	0.0233

4.2. Server Correlations and Latent Servers

We further studied the correlations among the six individual servers, and derived the relationship among the individual servers and the latent ones.

Table 2 shows the correlations among the six individual servers, which is calculated according to Eq. 3. Note that the matrix is not symmetric because $o_{i,j,l}$ is not always equal to $o_{j,i,l}$. As shown in Table 2, the correlation between two servers ranges from 0.25 to 0.59, which implies that some servers are more closely correlated than others in terms of contact prediction. Thus, traditional linear-regression-based consensus methods, which simply apply "majority voting" rule and assume the error is under a normal distribution, will fail when correct contacts are not supported by majority of individual servers.

Table 2. Correlations among the six individual servers. FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SPK: SPARK3.

Server	FDP	MGTH	RAP	FUG	SAM	SPK
FDP	1	0.344	0.426	0.250	0.296	0.410
MGTH	0.347	1	0.418	0.263	0.295	0.413
RAP	0.428	0.414	1	0.296	0.346	0.514
FUG	0.345	0.348	0.402	1	0.365	0.398
SAM	0.502	0.500	0.593	0.466	1	0.593
SPK	0.403	0.407	0.500	0.293	0.336	1

We then derived the relationship between the latent servers and the individual ones. As shown in Table 3, different latent independent servers represent different individual servers; for example, H_1 represents the common characteristics shared by these individual servers because the weights of H_1 on these individual servers are about the same; H_2 differentiate FUGUE3 from other servers; H_3 represents FOLDpro by a large positive weight, and represents mGenThreader by a large negative weight. Based on the eigenvalues, H_4 was eliminated since the eigenvalue for H_4 is much smaller than others. Thus, H_4 can be considered as random noise.

Table 3. Relationship among the six individual servers and latent servers. FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SPK: SPARK3.

Server	H1	H2	H3	H4	H5	H6
FDP	0.371	-0.351	0.655	-0.549	0.014	-0.081
MGTH	0.372	-0.258	-0.752	-0.477	-0.004	-0.016
RAP	0.418	-0.225	0.035	0.364	0.265	0.755
FUG	0.373	0.821	0.039	-0.218	0.369	0.012
SAM	0.490	0.202	0.034	0.227	-0.814	-0.036
SPK	0.410	-0.207	-0.023	0.487	0.359	-0.649

We derived the optimal weights for the latent servers by cross validation on the four sets. Correlated mutation is considered to be another independent latent server, because it provides a target sequence-related probability for each contact candidate. CM is calculated as previously described in [19, 23]. Each time we trained our ILP model on three of these four sets, and got a set of optimal weights, based on which a new prediction server is derived, named as S_1^*, S_2^*, S_3^*, and S_4^*, respectively. In this paper, by saying server S^*, we mean server S_i^* on test set i ($i = 1, 2, 3, 4$). Table 4 shows the linear combination representation of S^* on the individual servers and correlated mutation. We can see the four sets of weights are very similar. Note here, a negative

weight implies that the corresponding server's contribution has been over-expressed by other individual servers which have correlation with it.

Table 4. The linear combination representation of S^* on six individual servers and correlated mutation. FDP: FOLDpro, MGTH: mGenThreader, RAP: RAPTOR, FUG: FUGUE3, SAM: SAM-T02, SPK: SPARK3.

S	FDP	MGTH	RAP	FUG	SAM	SPK	CM
S_1^*	0.292	-0.283	1.272	1.470	0.230	0.618	0.300
S_2^*	0.305	-0.274	1.346	1.346	0.217	0.578	0.370
S_3^*	0.383	-0.290	1.373	1.357	0.141	0.650	0.280
S_4^*	0.287	-0.440	1.292	1.386	0.123	0.558	0.230

4.3. CASP7 Evaluation

We first assessed our consensus server S^* by Receiver Operating Characteristic (ROC) plots. ROC curves can provide an intuitionistic way to examine the tradeoff between the ability of a classifier to correctly identify positive cases and the number of negative cases that are incorrectly classified. Fig 1 shows the performance comparison in terms of contact prediction for server S^* and the six individual servers on the four test sets determined by our cross validation.

As shown in Fig 1, server S^* performs better than any individual server on all the four test sets. For each server, the performance of this server on test set 1 is slightly better than that on the other three test sets, which means test set 1 is the easiest test set among those four. RAPTOR performs better than other individual servers on the first three test sets, while SPARK3 has the best performance on test set 4. There are clear performance differences between server S^* and the best individual server on test set 1, 2, and 4 when the false positive rate is below 0.3. However, the difference is not obvious on those three test sets when false positive rate is larger than 0.3. For test set 3, the hardest test set, the performance of S^* is much better than any individual server all the time. It is also noticeable that the curve of S^* is much smoother than individual servers.

We further evaluated the performance of S^* from average accuracy point of view. Table 5 shows the average accuracy and deviation of S^* and "majority voting" server on the four test sets when different numbers of top contacts are considered. Recall S^* generates a confidence score for each contact candidate, we can easily take the top contacts for each target after sorting all candidates by their confidence scores. We implemented "majority voting" server as follows: For each contact candidate of a target generated by the best models of the six individual servers, the best model of each individual server votes "Yes" (denoted as 1) or "No" (denoted as 0) to this candidate. The number of supporting servers for all candidates are then calculated and sorted, and different numbers of top candidates are taken. The accuracy is calculated on the top candidate sets.

As shown in Table 5, the average accuracy increases when the number of top contacts decreases, except for server S^* on test set 1, on which the accuracy for top $L/10$ contacts is slightly lower than that for top $L/5$ contacts. This is possible because $L/10$ is usually a small number (20-30 for most cases), and a few incorrectly predicted top contacts will influence the total accuracy significantly. The overall accuracy of S^* on all four test sets is at least 62%, and is always higher than "majority voting" server. For the top $L/5$ contacts, the accuracy of S^* is 73.41%, which is about 5% higher than "majority voting".

We drew Fig 2 to examine the prediction accuracy for the top $L/5$ contacts of S^* on each CASP7 target. It can be seen that the accuracy is higher than 80% on most targets. In fact, among the total 85 targets, S^* has accuracy 100% on 13 targets, above 90% on 38 targets, and above 80% on 57 targets, while the accuracy is below 40% for only 16 targets. Note that there are two targets, T0309 (free modeling target) and T0335 (template based modeling target), on which S^* has accuracy 0. We carefully looked into these two targets. Both targets are very short. The target sequences published by CASP7 for T0309 and T0335 have length 76 and 85, respectively. However, the experimentally determined length used by CASP7 to evaluate these two targets are only 62 and 36, respectively, which means some parts of the targets are not experimentally determinable or not accurate enough. Thus, $L/5$ is only 12 and 7 for these two targets. Besides, all six individual servers did poorly on contact prediction on them, which means we only have a few correct candidates among a large number of incorrect ones. This can explain the failure of S^* on T0309 and T0335.

To evaluate more carefully how much our consensus method can improve upon individual servers and the simple "majority voting" method, we divided all targets into three categories: easy (high accuracy), medium (template based modeling), and hard (free

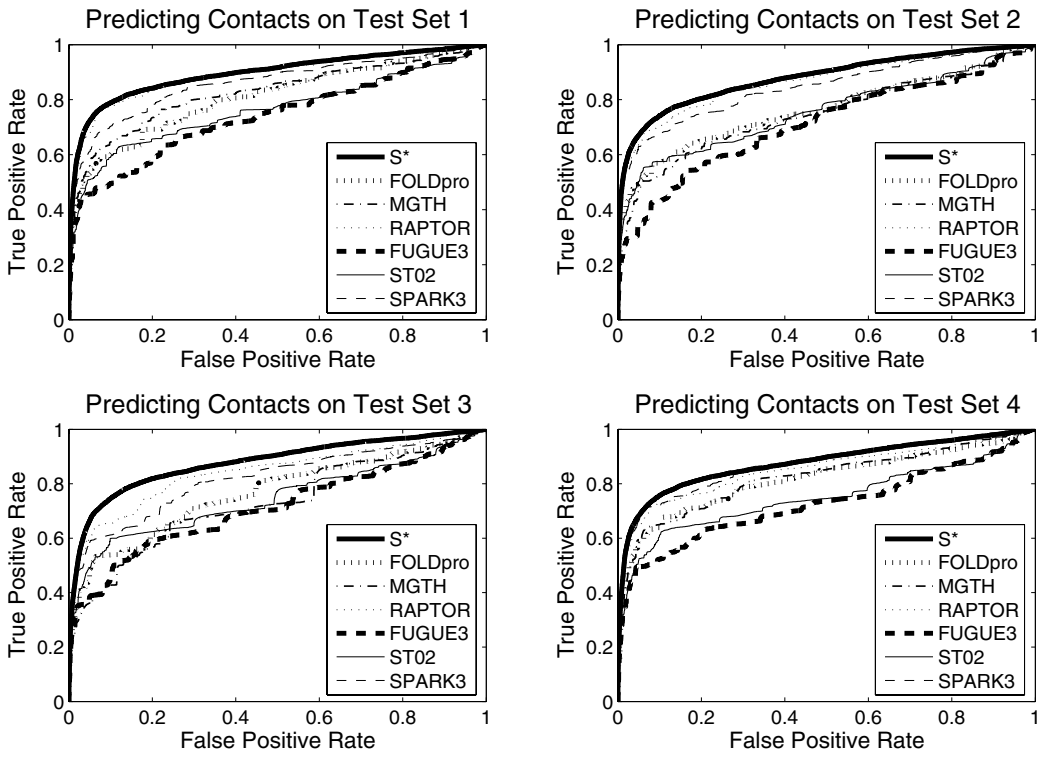

Fig. 1. Performance comparison using ROC plots in terms of contact prediction for S^* (thick solid line), FOLDpro (thick dotted line), mGenThreader (thin dashdot line), RAPTOR (thin dotted line), FUGUE3 (thick dashed line), SAM-T02 (thin solid line), and SPARK3 (thin dashed line).

Table 5. Average accuracy and deviation of the top contacts predicted by S^* on different test sets, and the accuracy of server "majority voting".

# Contacts	Test Set 1		Test Set 2		Test Set 3		Test Set 4		Overall		Majority Voting	
	Accu.	Dev.	Accu.	Dev.	Accu.	Dev.	Accu.	Dev.	Accu.	Dev.	Accu.	Dev.
L	0.6850	0.0815	0.6042	0.1116	0.5665	0.0860	0.6516	0.1239	**0.6267**	0.0986	**0.6093**	0.1064
$L/2$	0.7536	0.0857	0.6655	0.1080	0.6396	0.0964	0.7148	0.1310	**0.6932**	0.1029	**0.6600**	0.1122
$L/5$	0.8015	0.0843	0.7264	0.0909	0.6690	0.1081	0.7396	0.1240	**0.7341**	0.1001	**0.6872**	0.1130
$L/10$	0.7927	0.0965	0.7431	0.0799	0.6850	0.1083	0.7583	0.1218	**0.7445**	0.0994	**0.7048**	0.1149

modeling), according to Zhang's assessment [46]. The numbers of easy, medium and hard targets are 23, 46, and 16, respectively. Table 6 shows the average accuracy and deviation of S^*, individual servers, and "majority voting" method. As shown in Table 6, for easy and medium targets, the accuracy of S^* on top $L/5$ contacts is 93.71% and 75.85%, respectively, and much higher than the best individual server, where the improvement is at least 17% for each case. However, for hard targets, the accuracy of S^* is only 3% higher than SAM-T02, while at least 19% higher than the best of the rest servers. We examined the models generated by SAM-T02. They are sometimes much shorter than target proteins, and usually contain a very small set of contacts. However, the percentage of native contacts in this set is usually high. On the other hand, server S^* always performs better than "majority voting" server on easy, medium, and hard targets, while the improvements are about 2%, 4%, and 12%, respectively. This makes sense because for easy targets, individual servers usually do well, which means for a contact candidate, the more servers support it, the more likely it is correct. However, this rule doesn't always work on medium and hard targets. Thus, our consensus method does much better than "majority

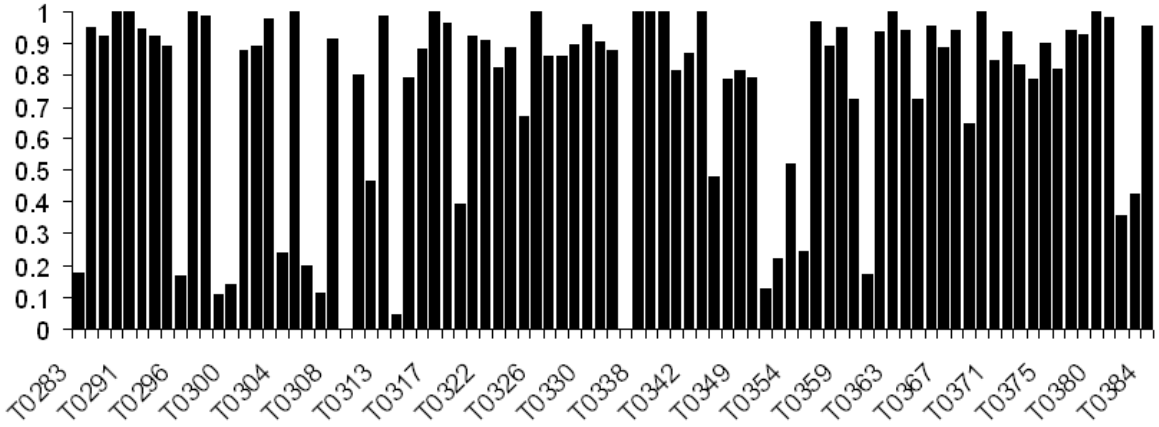

Fig. 2. Prediction accuracy for the top $L/5$ contacts of S^* on each CASP7 target.

Table 6. Average accuracy and deviation of S^*, individual servers, and "majority voting" server on easy, medium, and hard target sets

Server Name	Easy Targets		Medium Targets		Hard Targets	
	Accu.	Dev.	Accu.	Dev.	Accu.	Dev.
Top L of S^*	0.8957	0.0117	0.6378	0.0695	0.2083	0.0286
Top $L/2$ of S^*	0.9359	0.0060	0.7130	0.0776	0.2873	0.0675
Top $L/5$ of S^*	**0.9371**	0.0044	**0.7585**	0.0848	**0.3721**	0.0950
Top $L/10$ of S^*	0.9564	0.0059	0.7639	0.0891	0.4128	0.1057
FOLDpro	**0.7697**	0.0105	0.4401	0.0539	0.1358	0.0171
mGenThreader	0.6783	0.0379	0.4300	0.0440	0.1826	0.0245
RAPTOR	0.7534	0.0132	0.5000	0.0387	0.1607	0.0083
FUGUE3	0.7481	0.0067	0.4731	0.0616	0.1353	0.0127
SAM-T02	0.7538	0.0132	**0.5403**	0.0522	**0.3419**	0.0824
SPARK3	0.7621	0.0144	0.4843	0.0471	0.1707	0.0209
Top $L/5$ of Majority Voting	**0.9240**	0.0070	**0.7174**	0.0808	**0.2590**	0.0953

voting" on harder targets.

5. DISCUSSIONS

The experimental results have demonstrated that by encoding global energy and structure information from another perspective, consensus methods can identify native contacts well. We did not directly compare our method to other contact predictors on exactly the same data set and the same contact definition, since such data is not available. It is widely acknowledged that CASP data set is objective and comprehensive. Thus, it can be expected that our method performs much better than other predictors on the same data set because our method achieves an average accuracy of 73.41% on CASP7 data set, comparing to generally 30% accuracy of other predictors on data sets with similar difficulty levels to CASP.

One drawback of our method is that it is a selection-only consensus method. If all individual servers generate models with very few native contacts, our method will fail simply because there is nothing correct to select. We tried to avoid this drawback by using a server independent feature, CM, to introduce some contact candidates which are not predicted by any individual server. However, CM itself is not strong enough to find native contacts. Thus, future work will be combining more server independent features to introduce native contact candidates even if all individual servers fail to do so. On the other hand, a possibly better measure for consensus contact prediction methods is to require the methods to predict all the native contacts inside the input candidate set instead of predicting a fixed-size contact set. In this way, if all individual servers fail to predict any native contacts, and the consensus

method also returns 0 contacts, the accuracy will be 100%, which makes more sense than 0% under the current evaluation criteria.

A potential application of our contact prediction method is to provide highly conserved constraints for protein structure prediction or refinement methods. Recent CASPs show that fragment assembly based methods usually perform better than traditional comparative modeling methods, because instead of assuming there are known templates in the database which have similar structures to the targets, fragment assembly based methods only require some substructures with similar structures to some regions in templates. However, fragment assembly based methods usually suffer from huge search spaces. Our consensus method has an average accuracy 73.41% on top $L/5$ contacts, while for most cases, the accuracy is higher than 80%. Thus, our method can provide a reasonable number of highly conserved contacts for assembly step to significantly reduce the search space.

On the other hand, if all the individual servers we used predict the structure for a target protein extremely well or poorly, our consensus method will probably be able to only improve the assembly speed, rather than the accuracy. In the former case, since almost all contact candidates provided by these individual servers are correct ones, our method can only reduce the total number of well-conserved contacts and thus improve the speed for assembly step. In the latter case, since there are almost no correct contact candidates for our method to choose, the assembly accuracy can hardly benefit from our results. However, in any other cases, contacts provided by our method will greatly help assembly process. The reason is that our method can generate a small number of highly conserved contacts. Considering only a small number of contacts will reduce the assembly search space, and thus increase the speed. Moreover, experimental results have demonstrated that our method can generate contacts with higher accuracy than both contact predictors and protein structure prediction methods. This can reduce the risk of generating models with incorrect contacts, which will reduce the risk of selecting incorrect models from the final assembly decoy set, and thus will greatly increase the overall assembly accuracy.

6. CONCLUSIONS

In this paper, we proposed a linear programming based consensus contact prediction method. Experimental results show that this method preforms well, especially on easy and medium targets. The accuracy of our method is higher than any previously reported studies.

ACKNOWLEDGEMENT

This work is supported by NSERC Grant OGP0046506, and NSF of China Grant 60496324.

References

1. Y. Xu, D. Xu, and J. Liang. Computational Methods for Protein Structure Prediction and Modeling, 1st ed. *Springer* 2007.
2. Y. Xu, D. Xu, and J. Liang. Computational Methods for Protein Structure Prediction and Modeling, 2nd ed. *Springer* 2007.
3. J. Moult, T. Hubbard, K. Fidelis, and J. Pedersen. Critical assessment of methods of protein structure prediction (CASP):round III. *Proteins* 1999; **37**: 2–6.
4. J. Moult, K. Fidelis, A. Zemla, and T. Hubbard. Critical assessment of methods of protein structure prediction (CASP):round IV. *Proteins* 2001; **45**: 2–7.
5. J. Moult, K. Fidelis, A. Zemla, and T. Hubbard. Critical assessment of methods of protein structure prediction (CASP):round V. *Proteins* 2003; **53**: 334–339.
6. J. Moult, K. Fidelis, B. Rost, T. Hubbard, and A. Tramontano. Critical assessment of methods of protein structure prediction (CASP):round 6. *Proteins* 2005; **61**: 3–7.
7. O. Graña, D. Baker, R.M. MacCallum, J. Meiler, M. Punta, B. Rost, M.L. Tress, and A. Valencia. CASP6 assessment of contact prediction. *Proteins* 2005; **61**: 214–224.
8. N. Clarke, A. Valencia, J.M.G. Izarzugaza, M.L. Tress, and O. Graña. CASP7 assessment of contact prediction. *CASP7 presentation*, November 2006.
9. D. Chivian, D. E. Kim, L. Malmström, P. Bradley, T. Robertson, P. Murphy, C. E. Strauss, R. Bonneau, C. A. Rohl, and D. Baker. Automated prediction of CASP-5 structures using the Robetta server. *Proteins* 2003; **53(S6)**: 524–533.
10. D.E. Kim, D. Chivian, and D. Baker. Protein structure prediction and analysis using the Robetta server. *Nucleic Acids Research* 2004; **32**: 526–531.
11. D. Chivian, D.E. Kim, L. Malmström, J. Schonbrun, C. Rohl, and D. Baker. Prediction of CASP6 structures using automated Robetta protocols. *Proteins* 2005; **61(S7:1)**: 57–66.

12. K.M.S. Misura, D. Chivian, C.A. Rohl, D.E. Kim, and D. Baker. Physically realistic homology models built with Rosetta can be more accurate than their templates. *PNAS* 2006; **103**: 5361–5366.
13. Y. Zhang, A. Arakaki, and J. Skolnick. TASSER: An automated method for the prediction of protein tertiary structures in CASP6. *Proteins* 2005; **61(S7)**: 91–98.
14. Tress M. Valencia, A., I. Ezkurdia, G. López, and O. Graña. CASP6 assessment of contact prediction. *CASP6 presentation*, December 2004.
15. S. Miyazawa and R.L. Jernigan. Estimation of effective interresidue contact energies from protein crystal-structures quasi-chemical approximation. *Macromolecules* 1985; **18**: 534–552.
16. M.J. Sippl. Calculation of conformational ensembles from potentials of mean force. *J. Mol. Biol.* 1990; **213**: 859–883.
17. T. Grossman, R.M. Farber, and A.S. Lapedes. Neural Net Representations of Empirical Protein Potentials. In *Intelligent Systems in Molecular Biology* 1995; 154–161.
18. E.S. Huang, S. Subbiah, and M. Levitt. Recognizing native folds by the arrangement of hydrophobic and polar residues. *J. Mol. Biol* 1995; **249**: 493–507.
19. U. Göbel, C. Sander, R. Schneider, and A. Valencia. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function and Genetics* 1994; **18**: 309–317.
20. I.N. Shindyalov, N.A. Kolchanov, and C. Sander. Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Prot. Engng.* 1994; **7**: 349–358.
21. W.R. Taylor and K. Hatrick. Compensating changes in protein multiple sequence alignments. *Prot. Engng.* 1994; **7**: 341–348.
22. D.J. Thomas, G. Casari, and C. Sander. The prediction of protein contacts from multiple sequence alignments. *Prot. Engng.* 1996; **9**: 941–948.
23. O. Olmea and A. Valencia. Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Fold Des.* 1997; **2**: S25–32.
24. P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. *Proteins* 2001; **5**: 157–162.
25. P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Prediction of contact maps with neural networks and correlated mutations. *Protein Engineering* 2001; **14(11)**: 835–843.
26. M.S. Singer, G. Vriend, and R.P. Bywater. Prediction of protein residue contacts with a PDB-derived likelihood matrix. *Prot. Engng.* 2002; **15**: 721–725.
27. G. Pollastri and P. Baldi. Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. *Bioinformatics* 2002; **18**: S62–70.
28. Y. Zhao and G. Karypis. Prediction of contact maps using Support Vector Machines. In *3rd IEEE International Conference on Bioinformatics and Bioengineering (BIBE)* 2003; 26–33.
29. N. Hamilton, K. Burrage, M.A. Ragan, and T. Huber. Protein contact prediction using patterns of correlation. *Proteins: Structure, Function, and Bioinformatics* 2004; **56**: 679–684.
30. M. Punta and B. Rost. PROFcon: novel prediction of long-range contacts. *Bioinformatics* 2005; **21**: 2960–2968.
31. M.J. Zaki, S. Jin, and C. Bystroff. Mining residue contacts in proteins using local structure predictions. *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics* 2003; **33**: 789–801.
32. Y. Shao and C. Bystroff. Predicting interresidue contacts using templates and pathways. *Proteins* 2003; **53**: 497–502.
33. C. Bystroff, V. Thorsson, and D. Baker. HMMSTR: A hidden Markov model for local sequence-structure correlations in proteins. *J. Mol. Biol.* 2000; **301**: 173–190.
34. R.M. MacCallum. Striped sheets and protein contact prediction. *Bioinformatics* 2004; **20**: 224–231.
35. S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research* 1997; **25**: 3389–3402.
36. DT. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J.Mol.Biol.* 1999; **292**: 195–202.
37. J. Cheng and P. Baldi. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics* 2006; **22**: 1456–1463.
38. D.T. Jones. GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.* 1999; **287**: 797–815.
39. L.J. McGuffin and D.T. Jones. Improvement of the GenTHREADER method for genomic fold recognition. *Bioinformatics* 2003; **19**: 874–881.
40. J. Xu, M. Li, G. Lin, D. Kim, and Y. Xu. Protein threading by linear programming. In *Proceedings of the Pacific Symposium(PSB)* 2003; 264–275.
41. J. Xu. Protein fold recognition by predicted alignment accuracy. *ACM/IEEE Transactions on Computational Biology and Bioinformatics* 2005; **2(2)**: 157–165.
42. J. Shi, T.L. Blundell, and K. Mizuguchi. FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *J. Mol. Biol.* 2001; **310(1)**: 243–257.
43. Barrett C. Karplus, K. and R. Hughey. Hidden Markov Models for detecting remote protein homologies. *Bioinformatics* 1998; **14(10)**: 846–856.
44. H. Zhou and Y. Zhou. Folg recognition by combining sequence profiles derived from evolution and from

depth dependent structural alignment of fragments . *Proteins* 2005; **58**: 321–328.
45. W. Li and A. Godzik. Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 2006; **22**: 1658–1659.
46. http://zhang.bioinformatics.ku.edu/casp7/.